WECN Structural Embedding/Removal

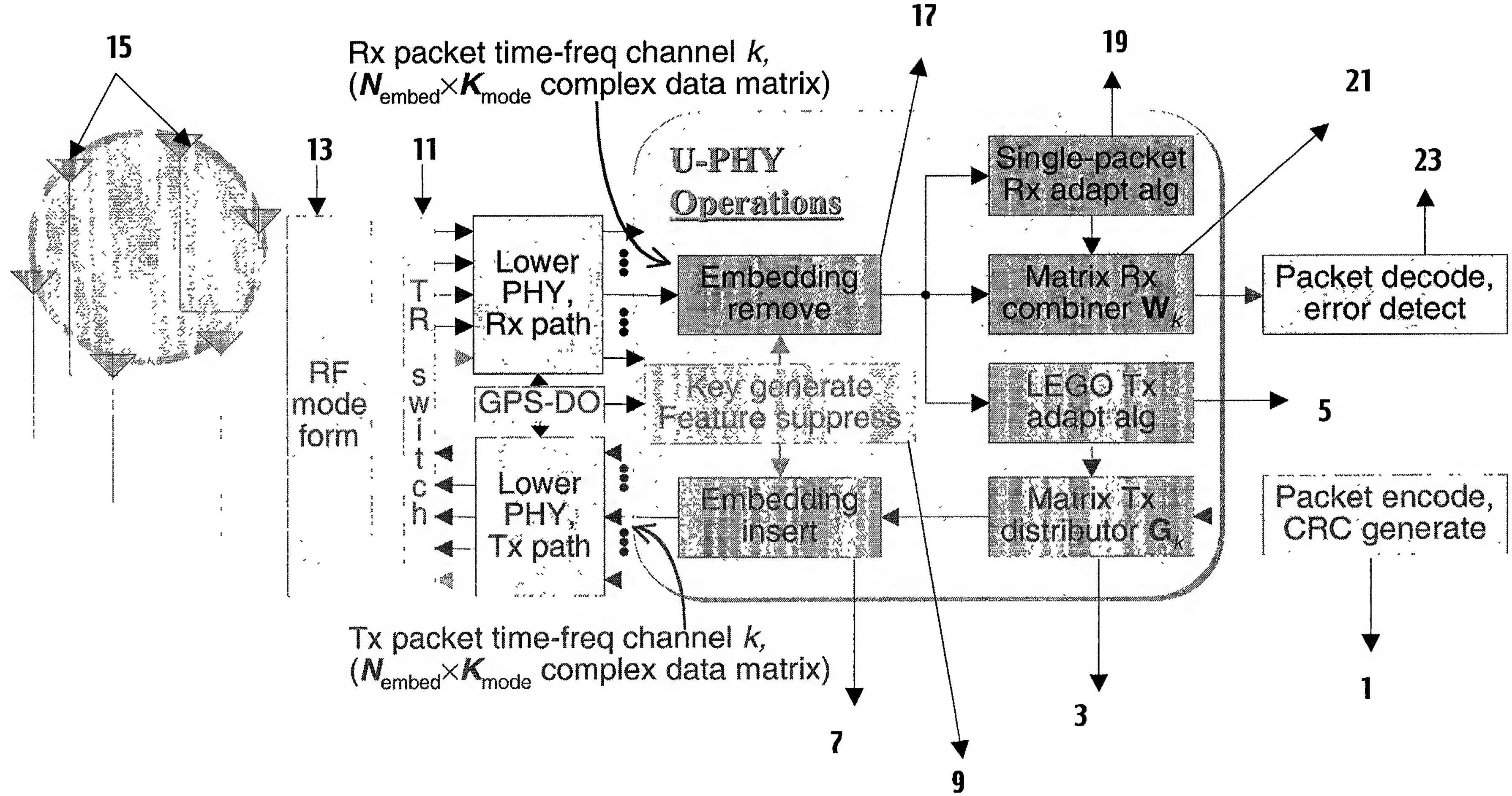


Figure 1

WECN with external sources/recipients

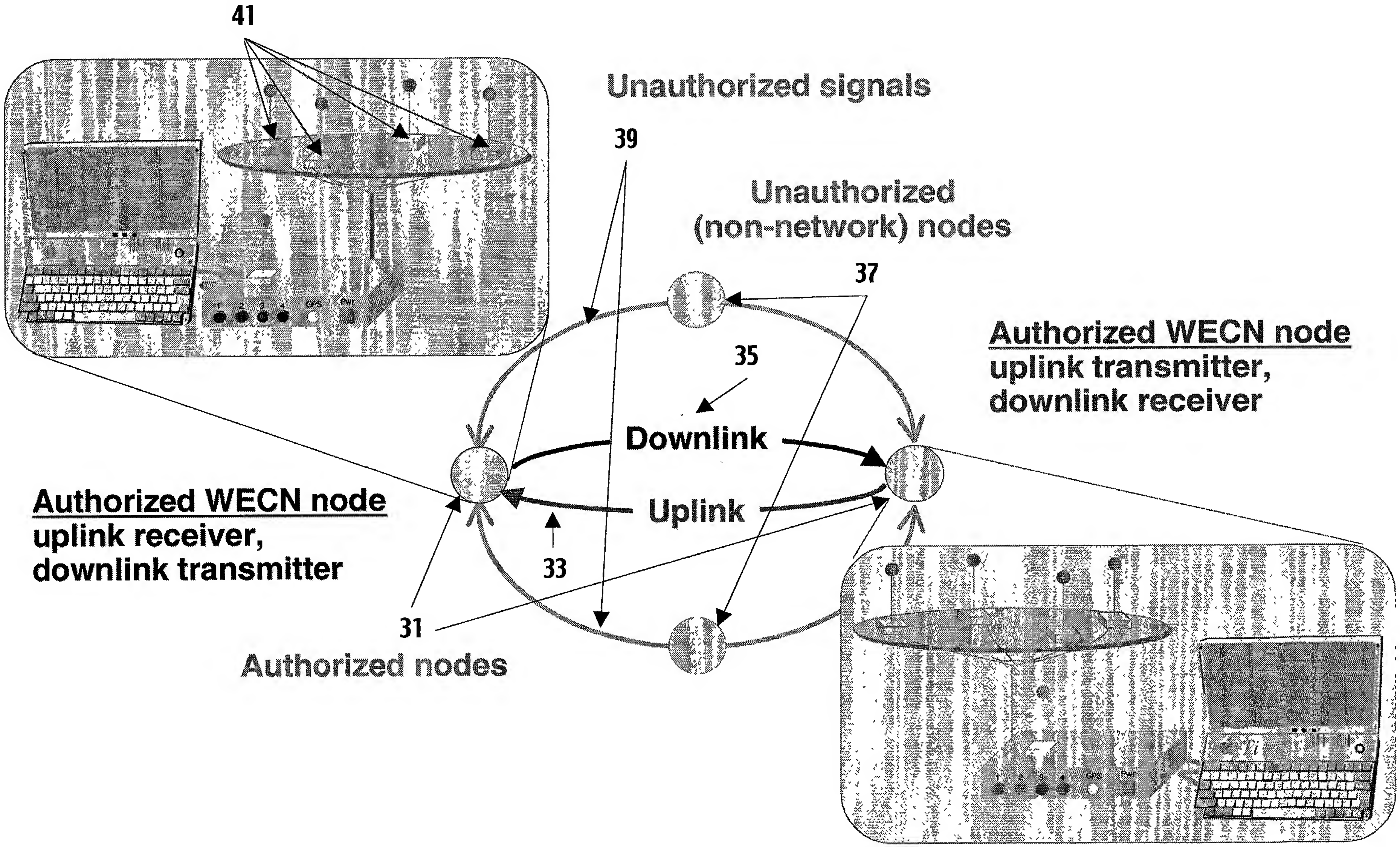


Figure 2

Time-Slot Embedding

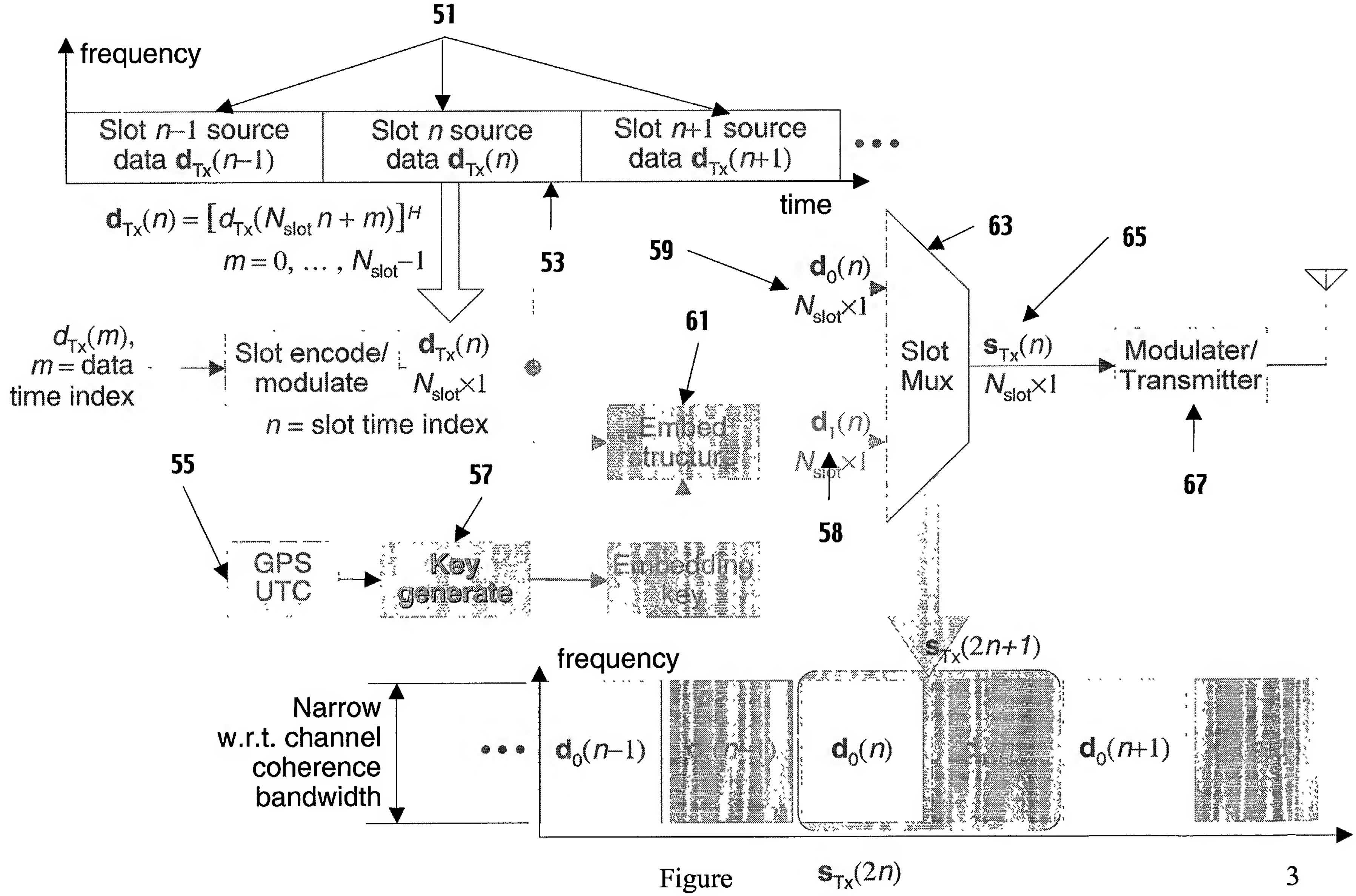


Figure 3

Time-Slot Reception/Removal

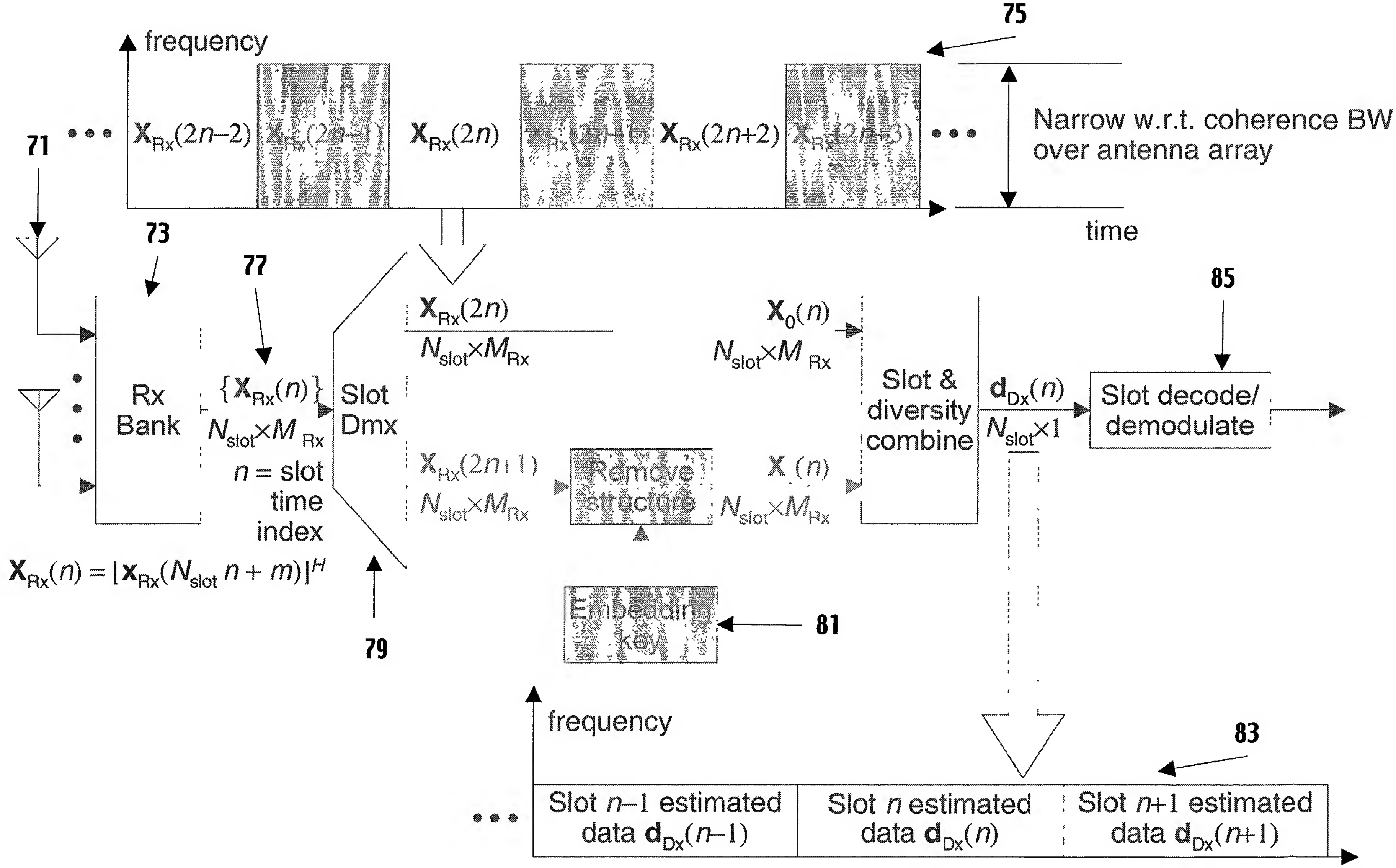


Figure 4

Delay-Invariant Embedding

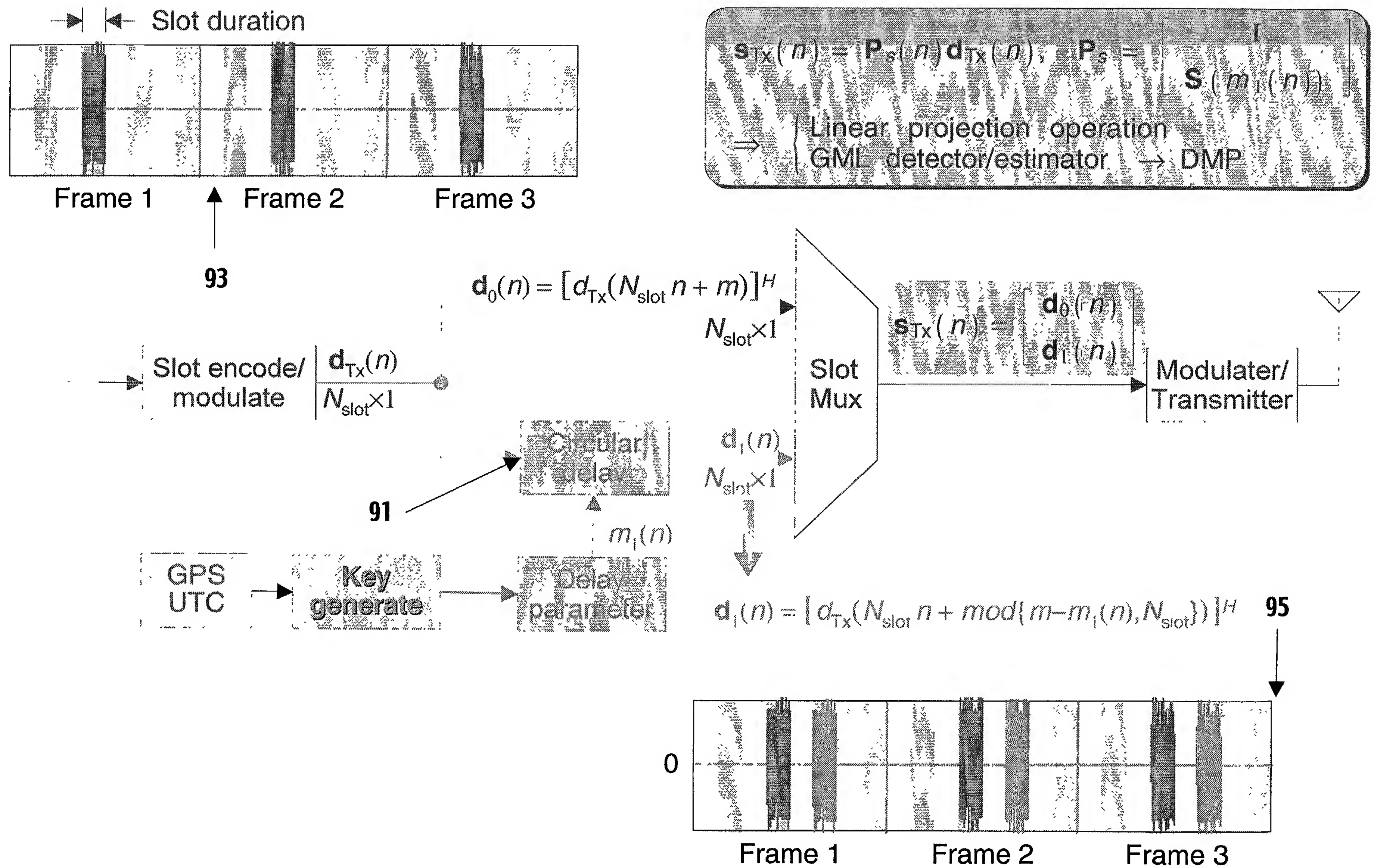


Figure 5

Frequency-Invariant Embedding

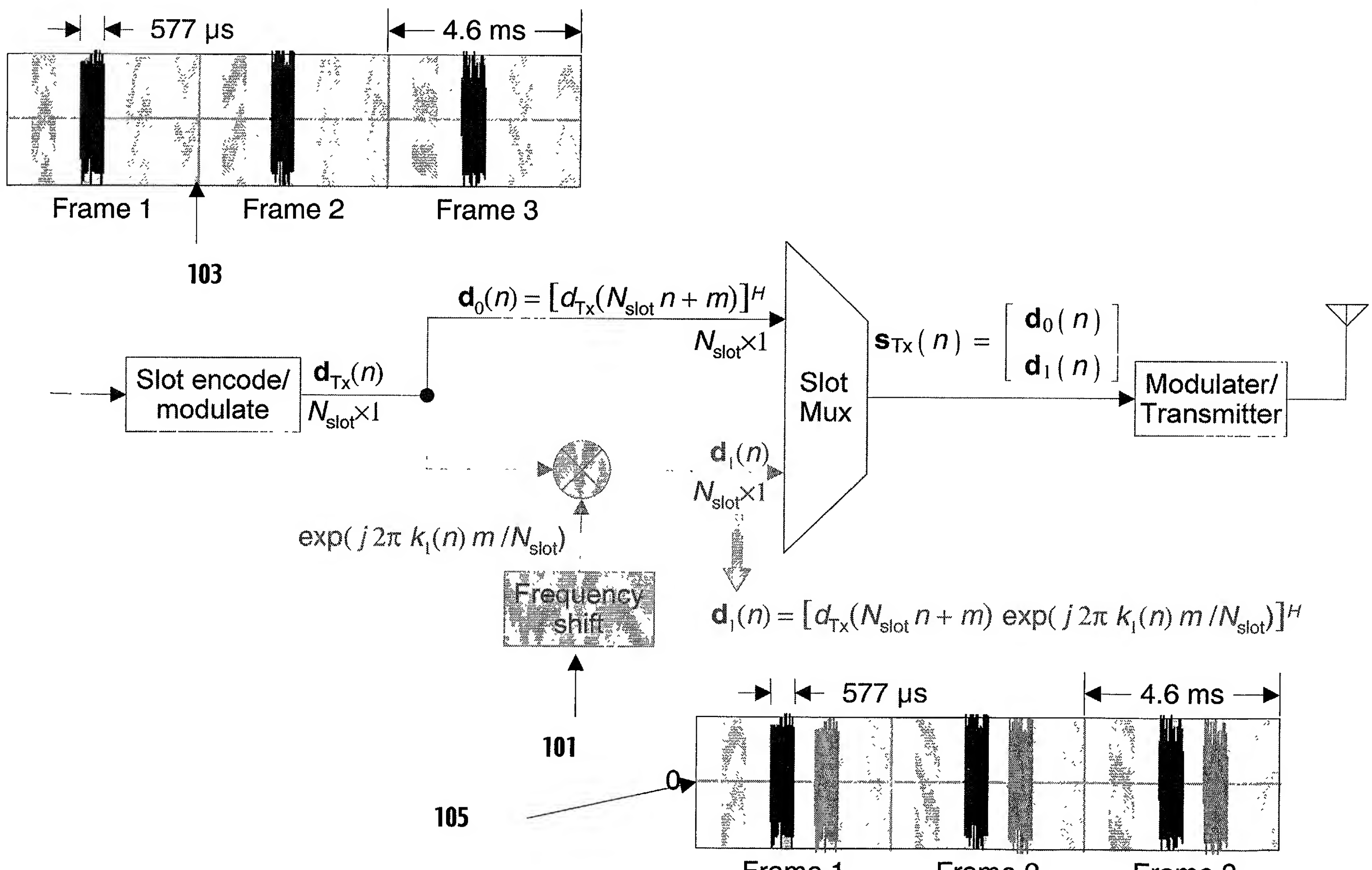
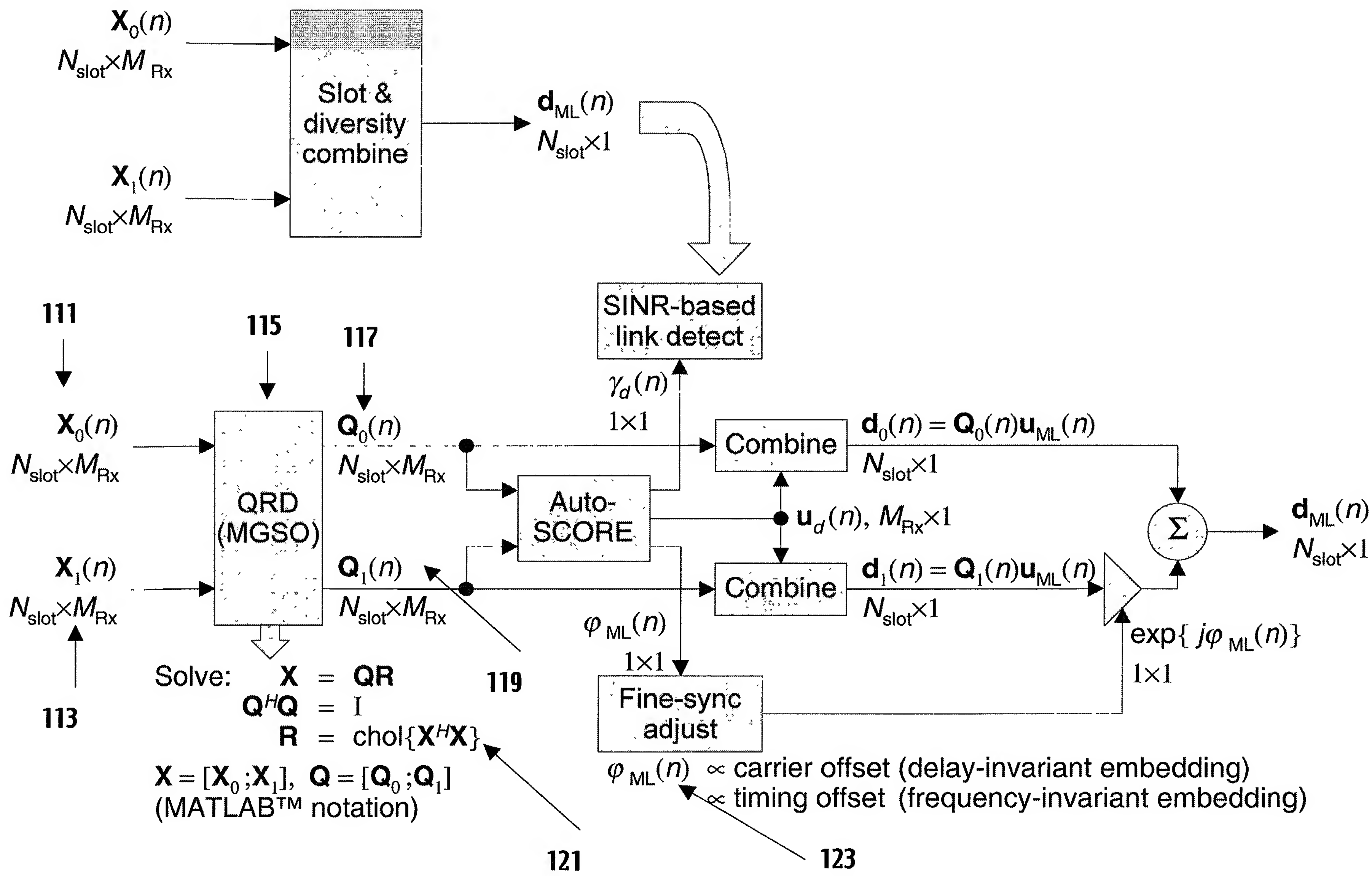


Figure 6

Auto-SCORE Adaptation - Data whitening & environmental evaluation



Figure

Single-Link Auto-SCORE Algorithm - Software

Combine $N_t \times M$ matrices $\mathbf{X}_0$ and $\mathbf{X}_1$ into single $2N \times M$ matrix $\mathbf{X}$,

$$\mathbf{X} = [\mathbf{X}_0^H \ \mathbf{X}_1^H]^H$$

where $N = N_{slot}$, $M = M_{Rx}$ if time-slot embedding is employed at the transmitter. Compute **QR** decomposition of $\mathbf{X}$,

$$\mathbf{X} = \mathbf{QR}$$
$$\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$$
$$\mathbf{R} = \mathrm{chol}\{\mathbf{X}^H\mathbf{X}\},$$

where $\mathbf{Q}$ is defined by

$$\mathbf{Q} = [\mathbf{q}_1 \ \dots \ \mathbf{q}_M] = [\mathbf{q}(1) \dots \mathbf{q}(N)]^H$$

Separate $\mathbf{Q}$ into $N \times M$ submatrices $\mathbf{Q}_0$ and $\mathbf{Q}_1$, such that

$$\mathbf{Q}_0 = \mathbf{X}_0\mathbf{C}$$
$$\mathbf{Q}_1 = \mathbf{X}_1\mathbf{C},$$

where $\mathbf{C} = \mathbf{R}^{-1}$. Form $M \times M$ cross-correlation matrix $\mathbf{S}$,

$$\mathbf{S} = (1/N)\, \mathbf{Q}_0^H \mathbf{Q}_1.$$

Initialize whitened linear combiner weights

$$\mathbf{u} = [s\,(m,M)]$$
$$\nu = \|\mathbf{u}\| \ (\mathrm{L}_2 \text{ norm})$$
$$\mathbf{u} \leftarrow \mathbf{u}/\nu$$

Iteratively update combiner weights (preset iterations, or until stopping criterion met)

$$\mathbf{v} = \mathbf{Su}$$
$$\rho = 1/2\ \mathrm{sign}\{\mathbf{v}^H\mathbf{u}\}$$
$$\mathbf{u} \leftarrow \rho\,\mathbf{v} + \rho^{*}\mathbf{S}^H\mathbf{u}$$
$$\nu = \|\mathbf{u}\| \ (\mathrm{L}_2 \text{ norm})$$
$$\mathbf{u} \leftarrow \mathbf{u}/\nu$$

Compute output SINR measurement γ,

$$\gamma = \nu/(1-\nu)$$

$\gamma \geq \gamma_{detect}$?

no → Exit

yes

Compute phase-shift estimate φ,

$$\varphi = \arg\{\rho\}$$

Compute slot/diversity combined output data

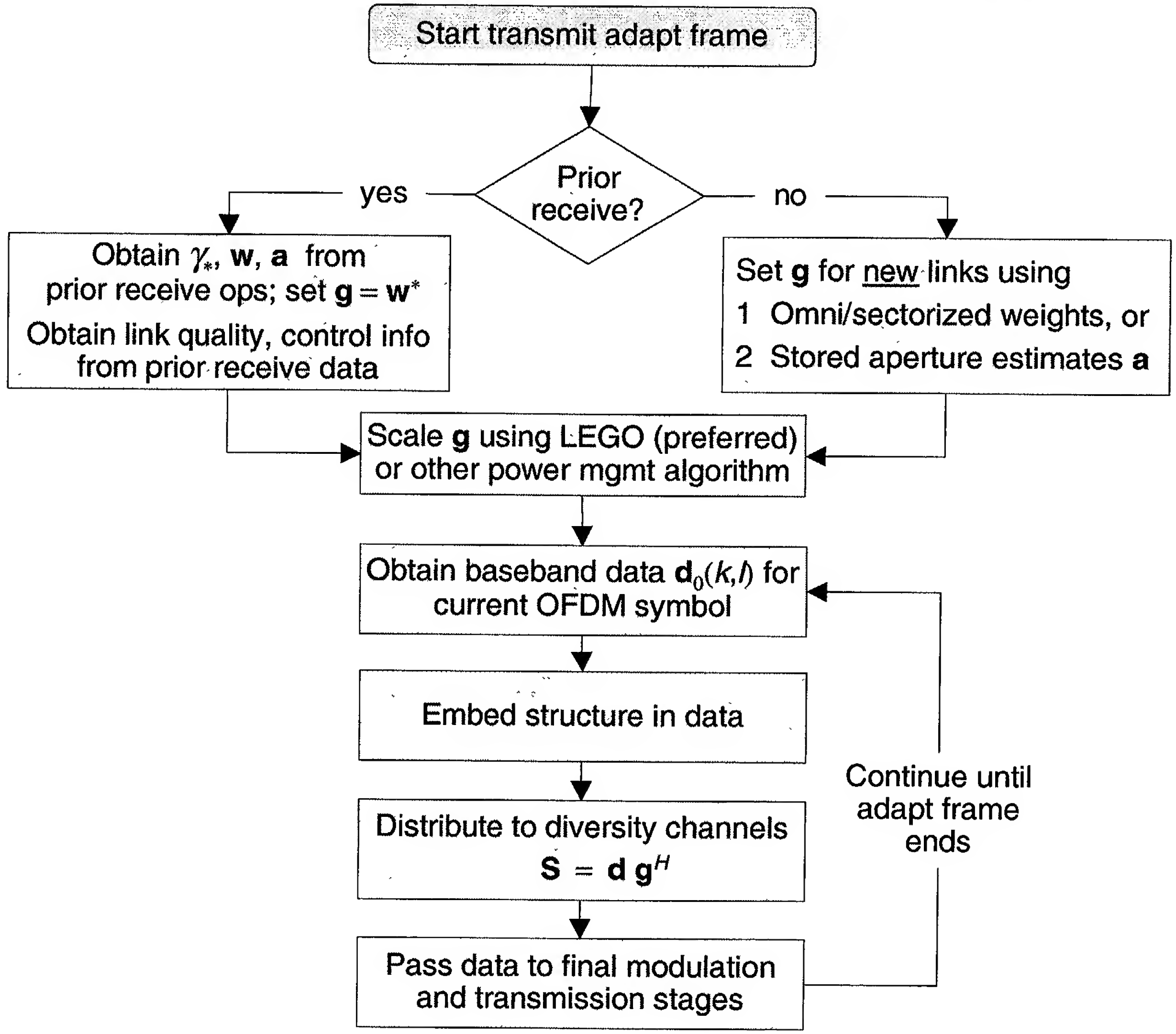
$$\mathbf{d} = \mathbf{Q}_0\mathbf{u} + (\mathbf{Q}_1\mathbf{u})e^{-j\varphi}$$

Compute unwhitened combiner weights $\mathbf{w}$, aperture vector $\mathbf{a}$,

$$\mathbf{w} = \mathbf{Cu}$$
$$\mathbf{a} = \mathbf{R}^H\mathbf{u}$$

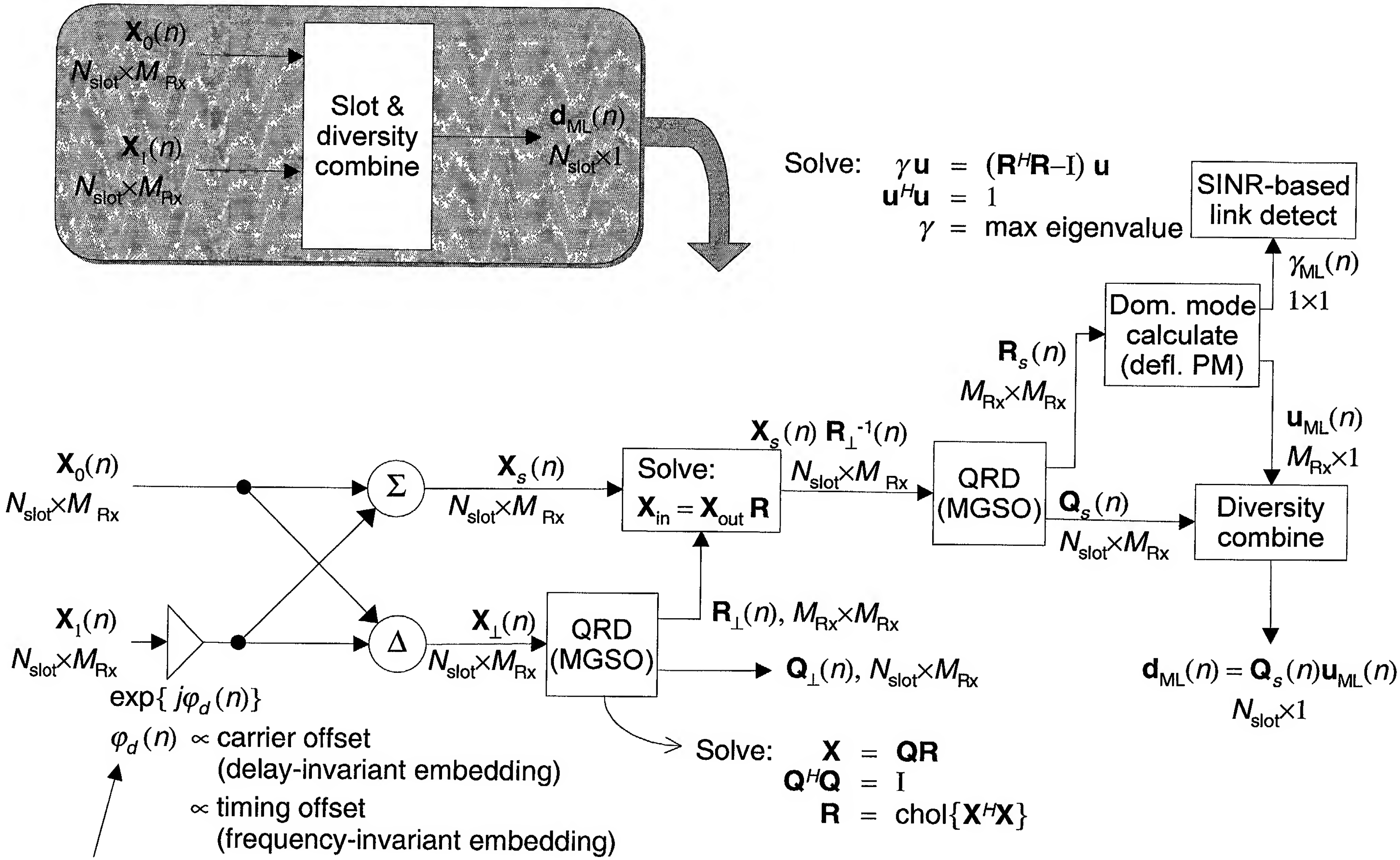
Figure

Single-Link Transmitter Flow Diagram



Figure

Alternative DMP Adaptation Algorithm



131

Figure

alternative converging embedded-signal-differentiation algorithms

Dominant-Mode Prediction

Solve:

$$\gamma \mathbf{u} = (\mathbf{R}^H\mathbf{R} - \mathrm{I})\, \mathbf{u}$$

$$\|\mathbf{u}\| = 1 \quad (\mathrm{L}_2 \text{ norm})$$

$$\gamma = \text{max eigenvalue}$$

Optimization Algorithm

Initialize:

$$\mathbf{u} = r(M,M)\, [r^*(M,1) - 1]$$

$$\gamma = \|\mathbf{u}\| \quad (\mathrm{L}_2 \text{ norm})$$

$$\mathbf{u} \leftarrow \mathbf{u}/\gamma$$

Iterate:

$$\mathbf{v} = \mathbf{R}\,\mathbf{u}$$

$$\mathbf{u} \leftarrow \mathbf{R}^H\mathbf{v} - \mathbf{u}$$

$$\gamma = \|\mathbf{u}\| \quad (\mathrm{L}_2 \text{ norm})$$

$$\mathbf{u} \leftarrow \mathbf{u}/\gamma$$

Auto-SCORE

Solve:

$$\nu(\varphi)\, \mathbf{u} = \mathbf{S}(\varphi)\, \mathbf{u}$$

$$\mathbf{S}(\varphi) = 1/2(\mathbf{S}e^{j\varphi} + \mathbf{S}^H e^{-j\varphi})$$

$$\|\mathbf{u}\| = 1 \quad (\mathrm{L}_2 \text{ norm})$$

$$\nu(\varphi) = \text{max eigenvalue}$$

$$\varphi = \arg\max_{\varphi} \nu(\varphi)$$

Optimization Algorithm

Initialize:

$$\mathbf{u} = [s(m,M)]$$

$$\nu = \|\mathbf{u}\| \quad (\mathrm{L}_2 \text{ norm})$$

$$\mathbf{u} \leftarrow \mathbf{u}/\nu$$

Iterate:

$$\mathbf{v} = \mathbf{S}\mathbf{u}$$

$$\rho = 1/2\ \mathrm{sign}\{\mathbf{v}^H\mathbf{u}\}$$

$$\mathbf{u} \leftarrow \rho\mathbf{v} + \rho^*\mathbf{S}^H\mathbf{u}$$

$$\nu = \|\mathbf{u}\| \quad (\mathrm{L}_2 \text{ norm})$$

$$\mathbf{u} \leftarrow \mathbf{u}/\nu$$

Finalize:

$$\varphi = \arg\{\rho\}$$

$$\gamma = \nu/(1-\nu)$$

Figure

10620T"T2S6E00T

Embedded Invariance Flowchart (Nodal View)

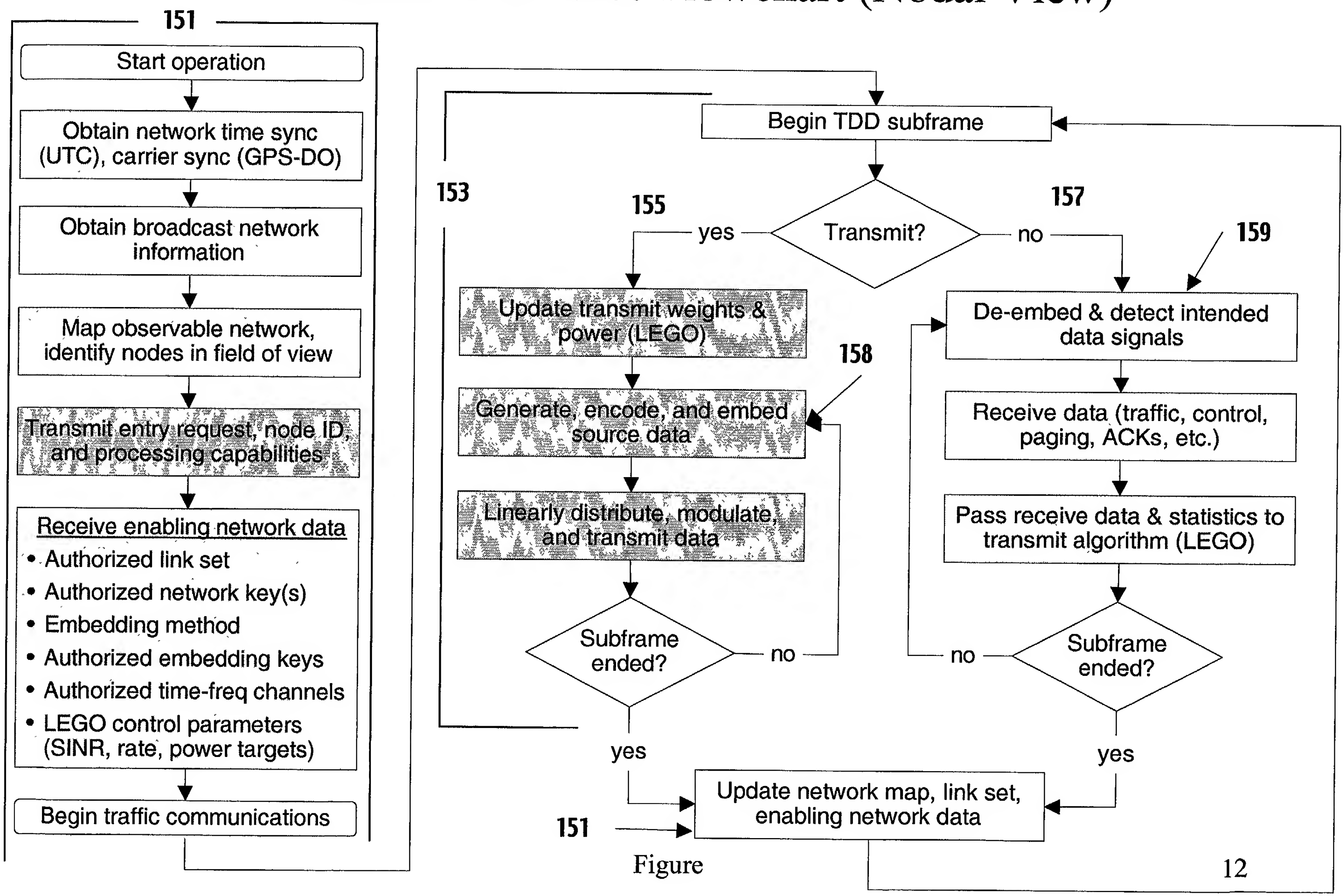


Figure 12

Embedded Invariance Flowchart (Link View)

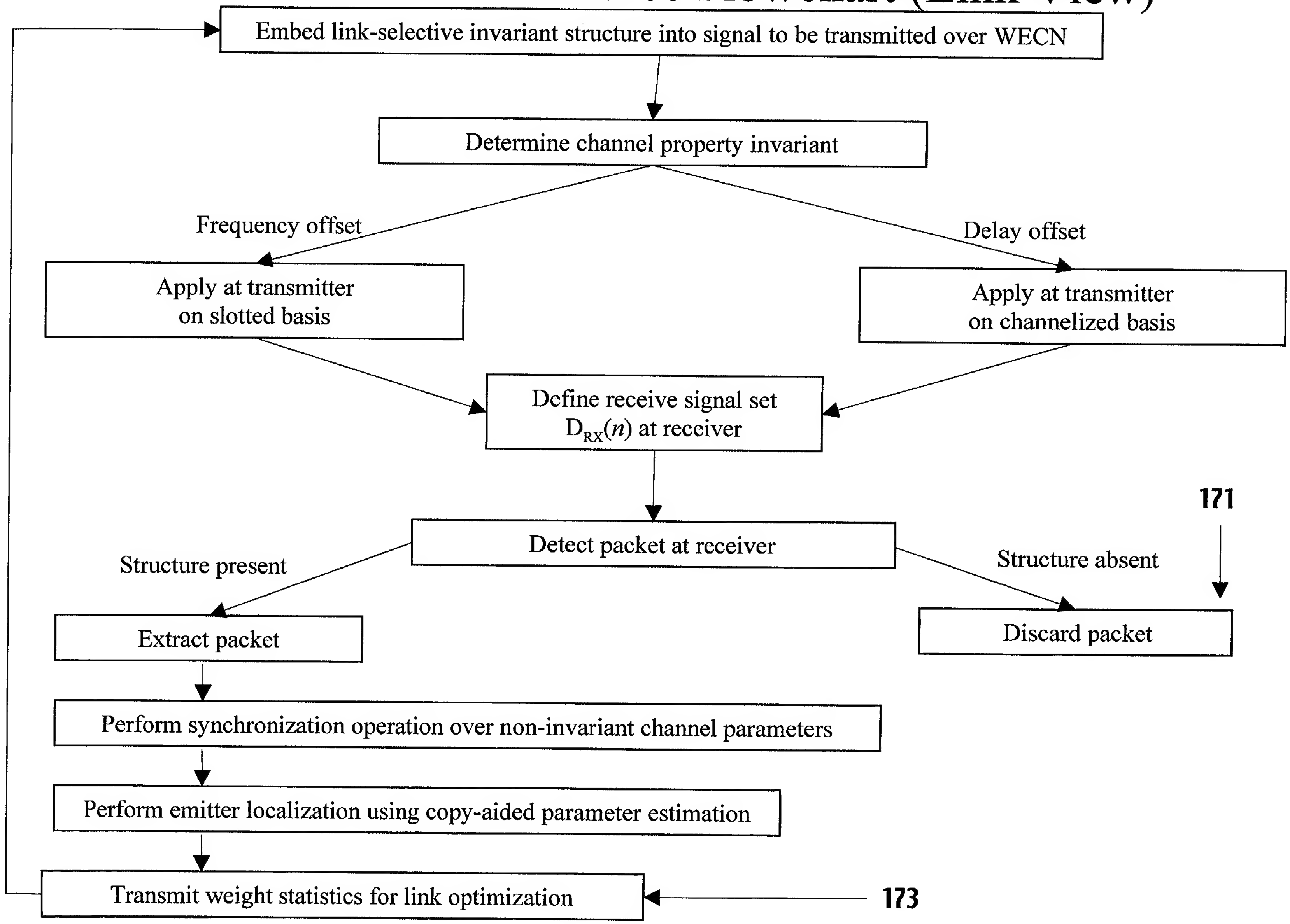
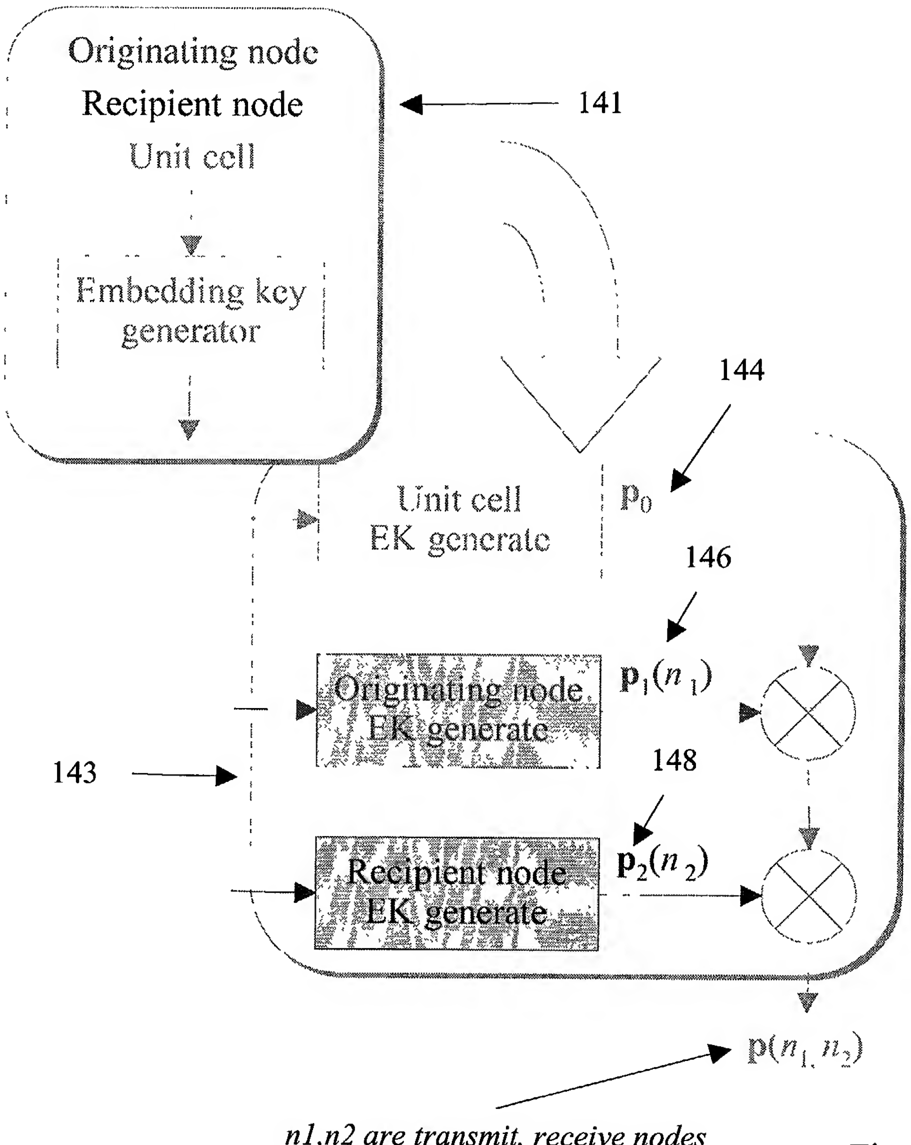


Figure 13

Multilink Embedding Key Generation Algorithm



$\mathbf{p}_0 = [c_0(k)]_{k=0,\ldots,K-1}$

K – Number tones used by link

$c_0(k)$ – Pseudorandom sequence, shared by all nodes in same subnet (can vary over tones and symbols)

$\mathbf{p}_1(n_1) = [p_1(k;n_1)]_{k=0,\ldots,K-1}$

$p_1(k;n_1) = \exp[-j\, 2\pi k\, m(n_1)/L]$

$m(n_1)$ = node <u>pseudodelay</u>, unique to (transmit) node n_1 in the subnet (varied over symbols to improve security)

$\mathbf{p}_2(n_2) = [c_2(k;n_2)]_{k=0,\ldots,K-1}$

$c_2(k;n_2)$ = Pseudorandom sequence, unique to (receive) node n_2 in subnet (varied over symbols to improve security)

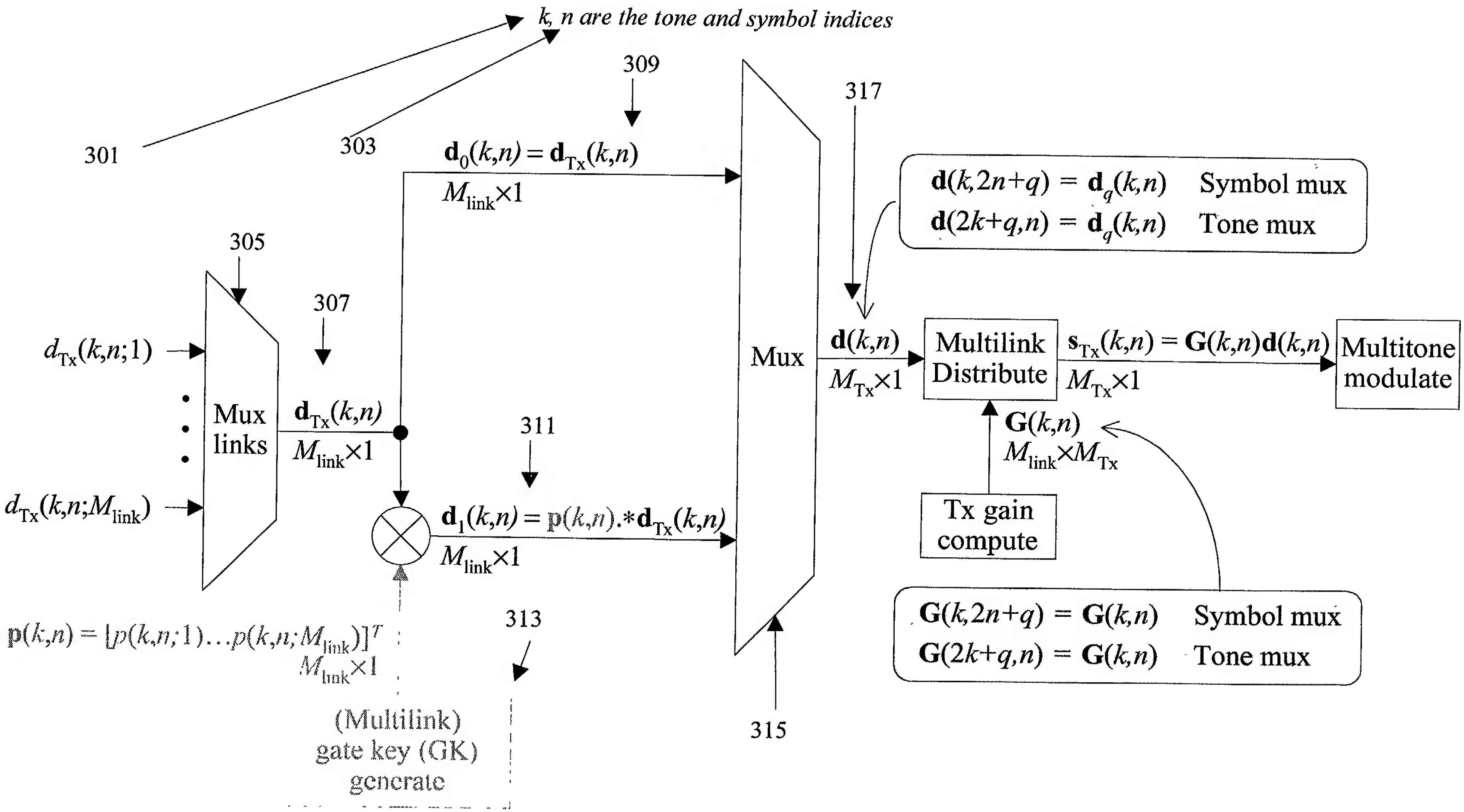
145

$\mathbf{p}(n_1, n_2) = \mathbf{p}_0 .* \mathbf{p}_1(n_1) .* \mathbf{p}_2(n_2)$

$= [c_0(k) p_1(k;n_1) c_2(k;n_2)]_{k=0,\ldots,K-1}$

Figure

Multilink Transmit Embedding Hardware (Node n_1)



Figure

Multilink Embedding Flow Diagram

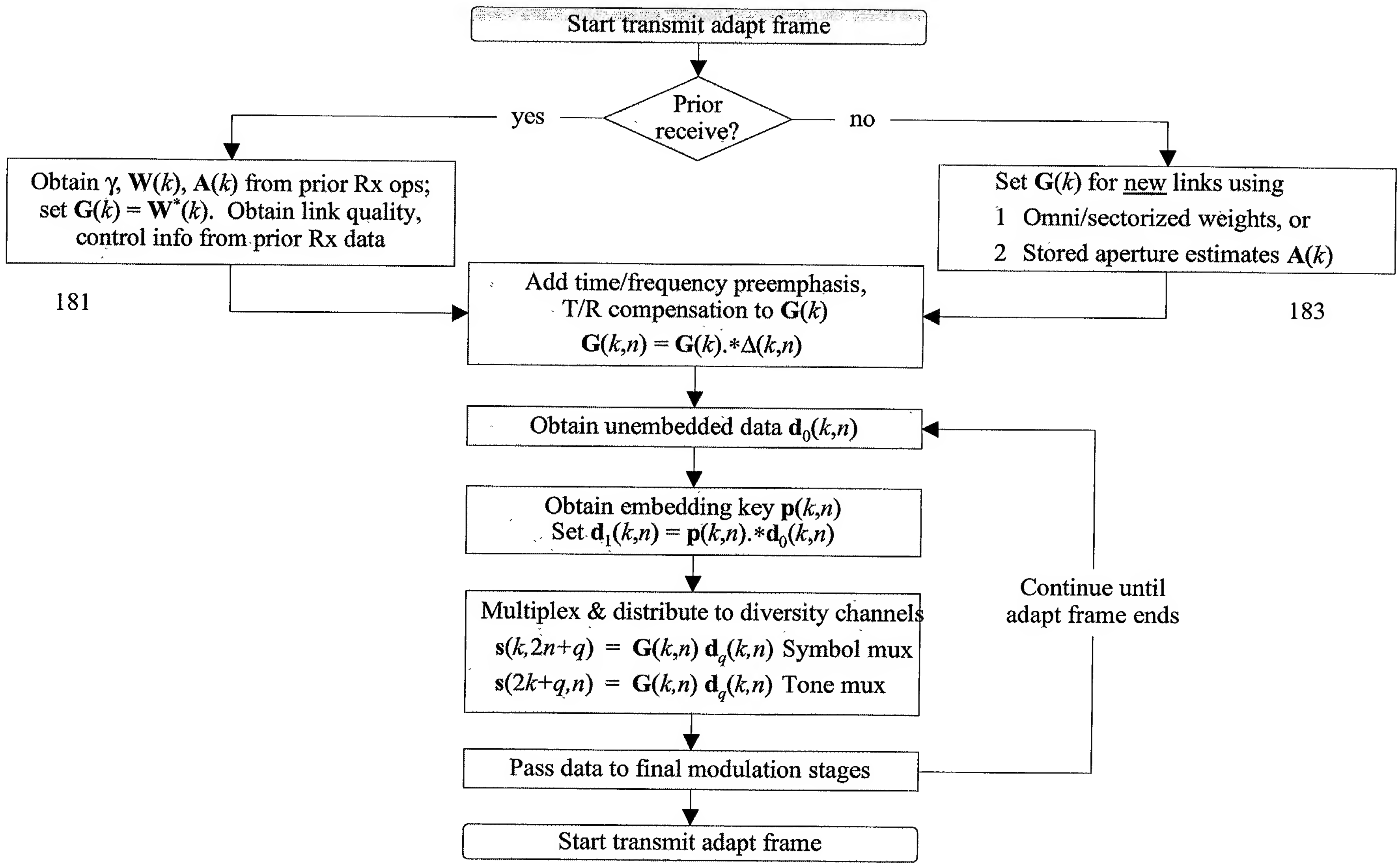
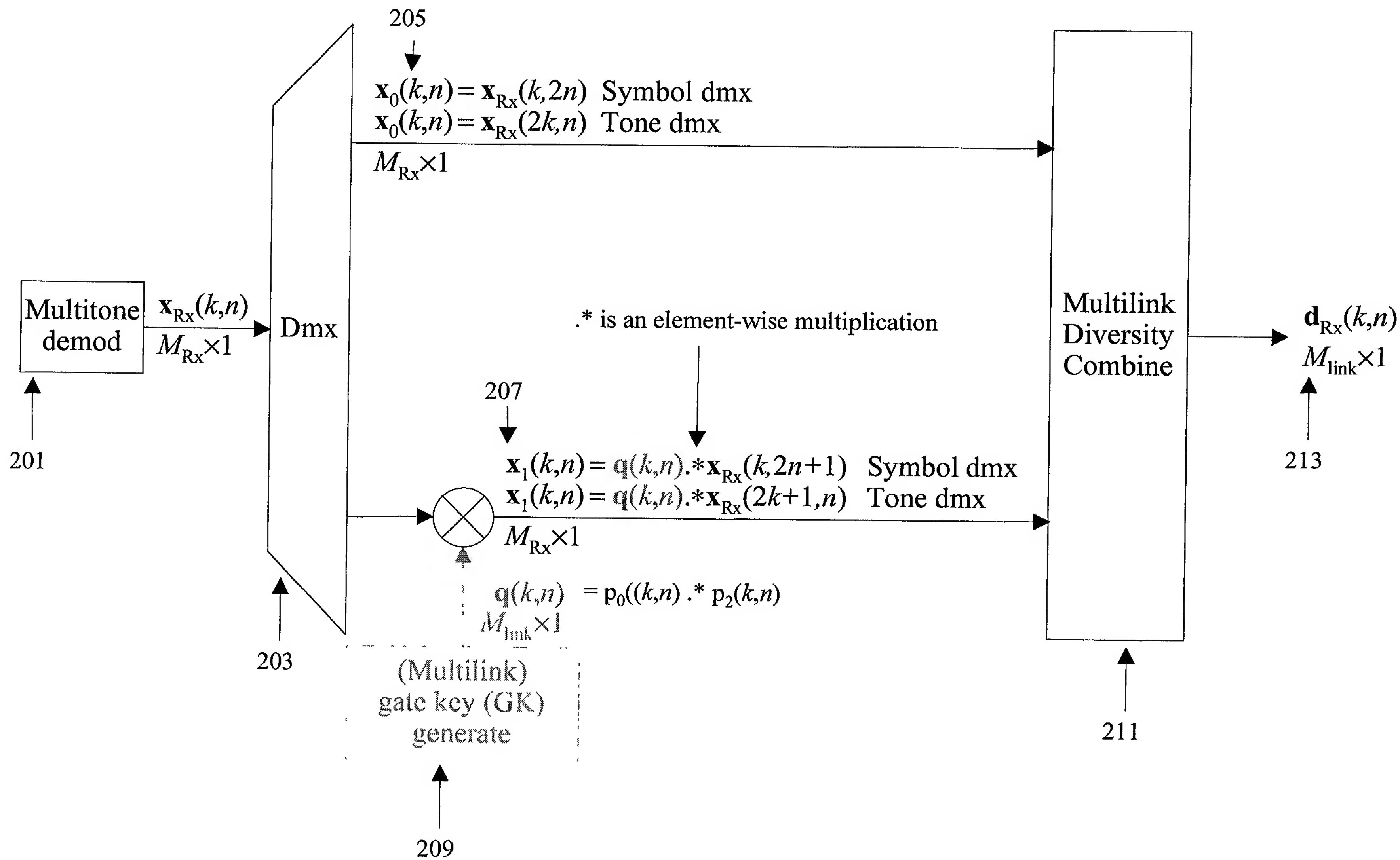


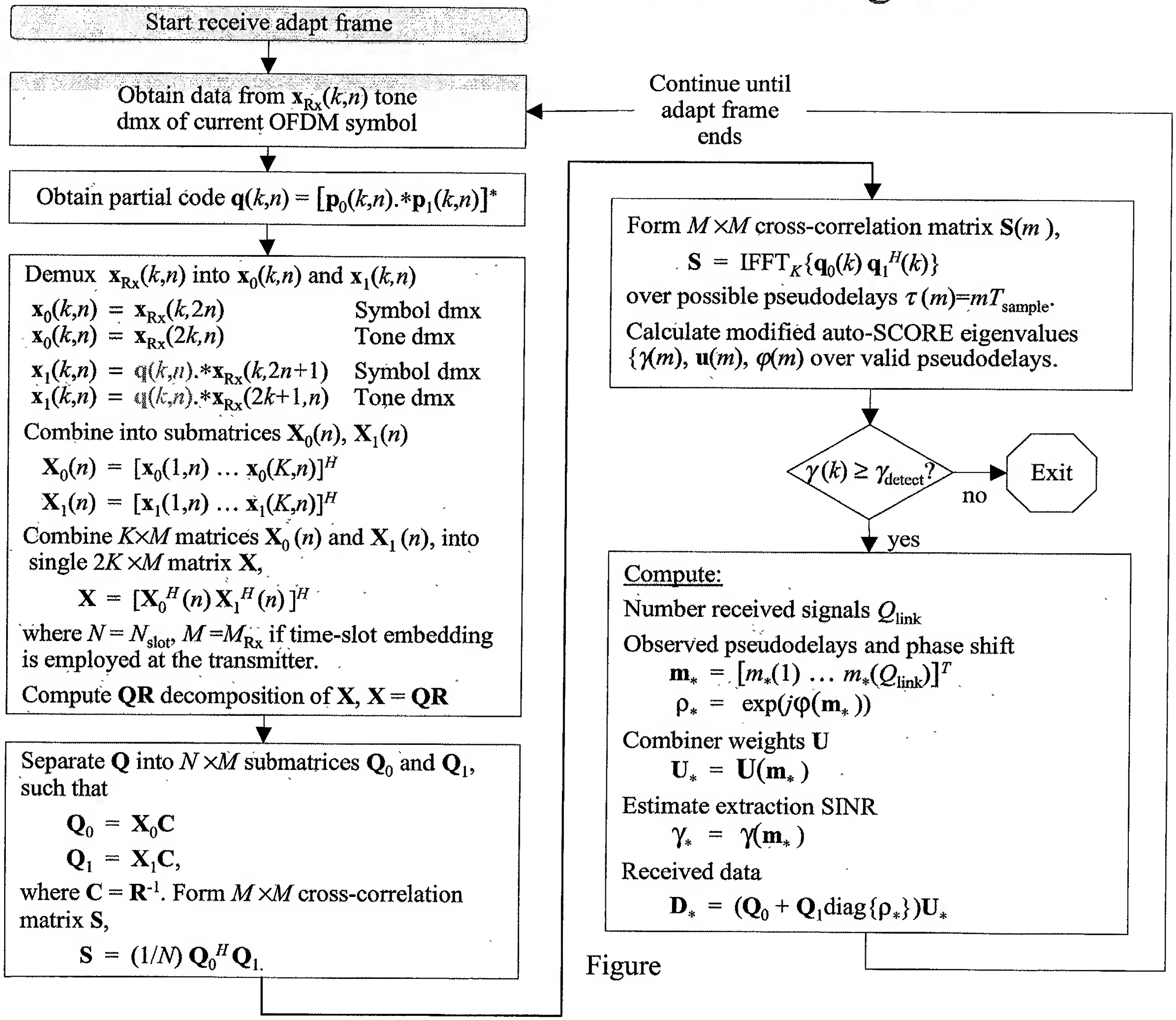
Figure 16

Multilink Receive Embedding Hardware



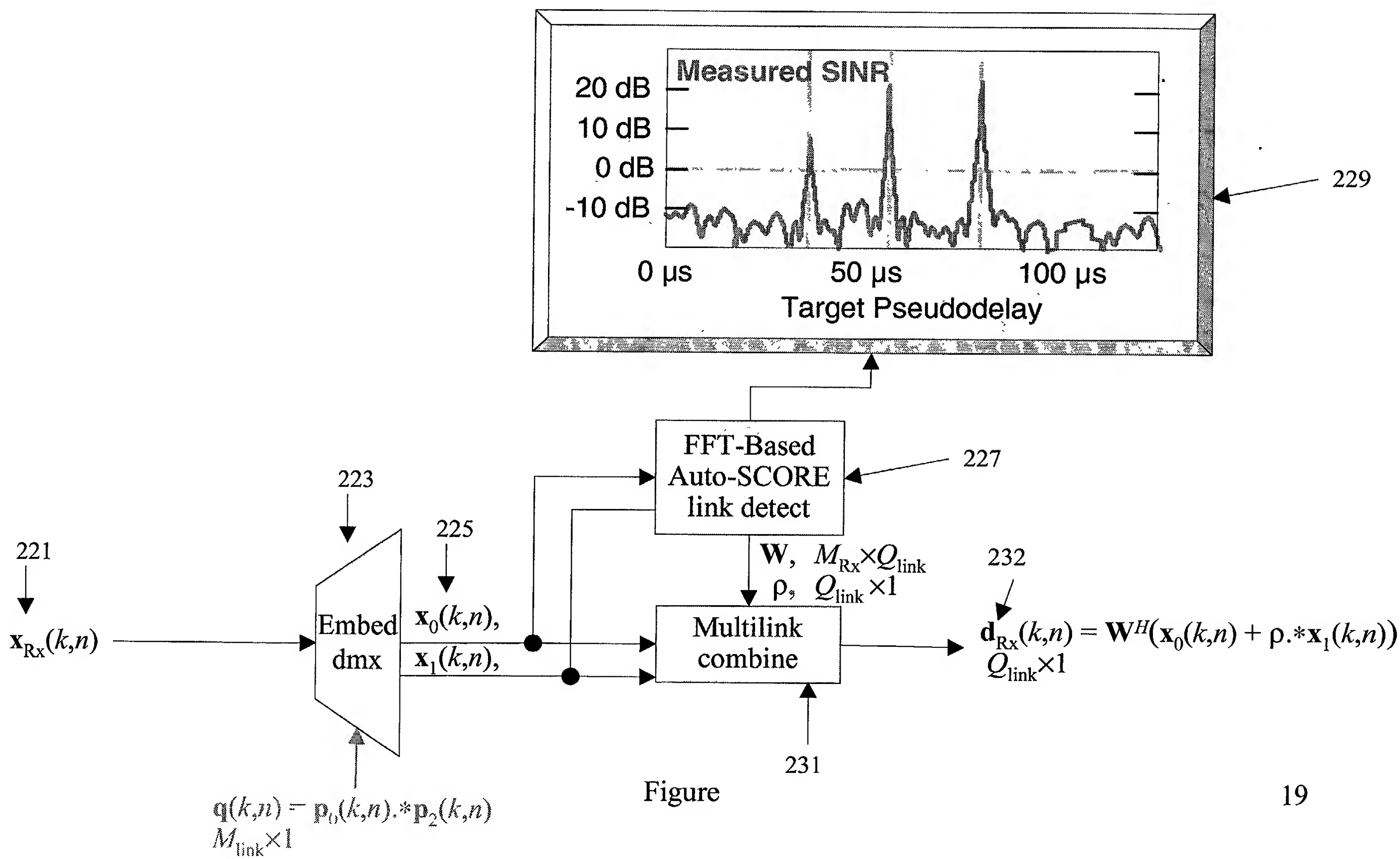
Figure

Multilink Receiver Flow Diagram



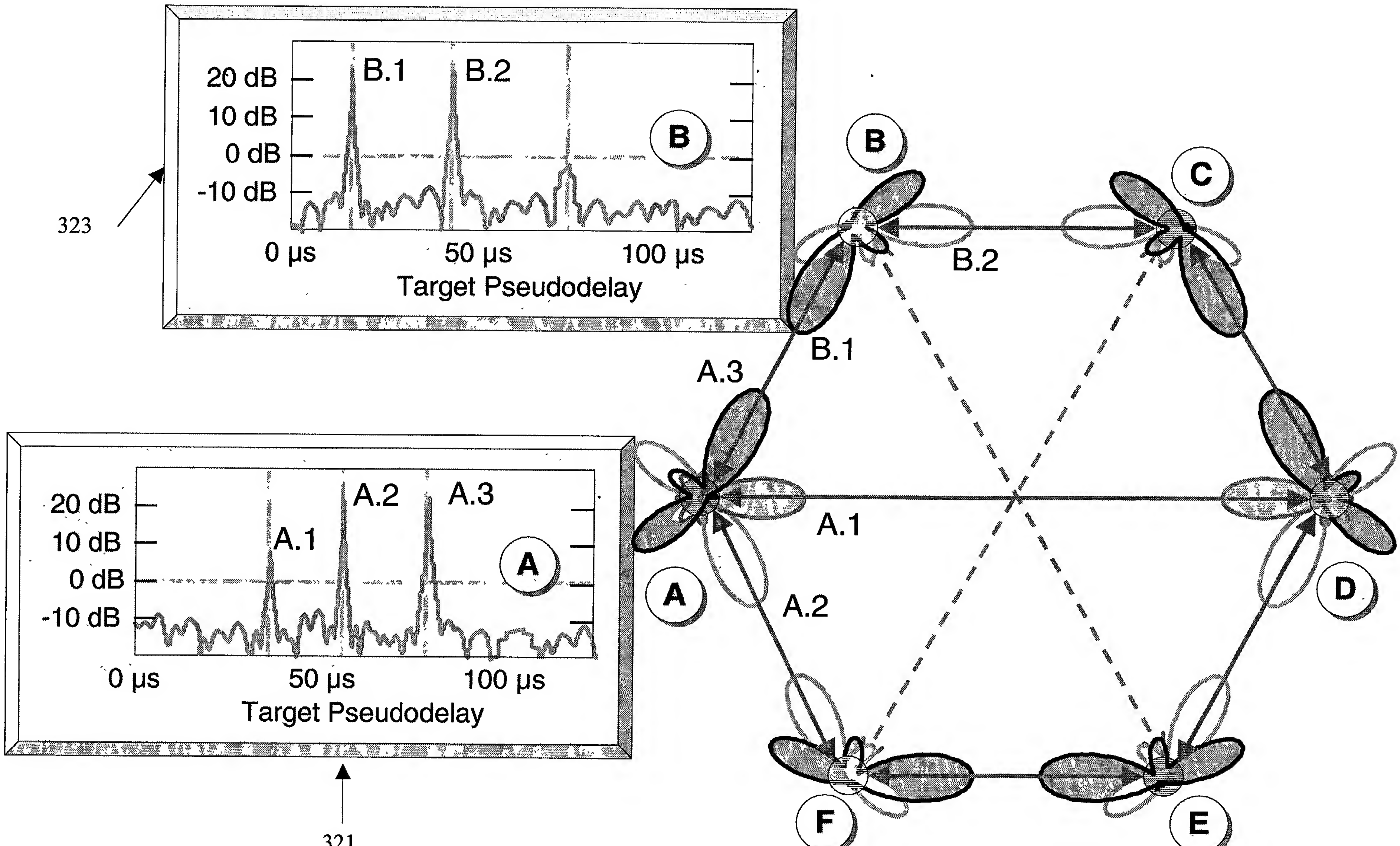
Figure

Link Detection, Separation Operation



Figure

Pseudodelay Plots and Antenna Patterns



Figure